Amendments to the Drawings

The attached 1 sheet of drawings includes changes to Figure 14. This sheet replaces the original 1 sheet of drawings.

Attachment: Replacement Figure 14

Remarks

Applicants respectfully request reconsideration of the present application in view of the foregoing amendments and the following remarks. Claims 1-20 are pending in the application. Claims 1-20 are rejected. Claim 17 has been canceled without prejudice. No claims have been allowed. Claims 1, 7, and 15 are independent.

Requirement for Information/Duty to Disclose

The Examiner has requested that the applicants make seven documents of record. In an IDS being filed with this Amendment, applicants are providing six of the references. One of the references requested, Reference 4.6 – Grieskamp et al., *Testing with Abstract State Machines* in Roberto Moreno-Diaz and Alexis Quesada-Arencibia, Formal Methods and Tools for Computer Science – EUROCAST'01 – Extended Abstracts, February 2001 – is not readily available to the applicants, and so is not being included in the IDS.

Amendments to the Drawings

The Examiner has objected to Figure 14, as, in his characterization, "only what which is old is illustrated." [Action, ¶6.] However, Figure 14, as recited by its description in the "Brief Description of the Drawings" is "a block diagram of a distributed computer system implementing the described technologies." This is reinforced by the description of block 1456, which initially was said to be a program module with "an implementation." [Application, page 27, line 25.] With this Amendment, the description of the implementation 1456 has been amended to read "an implementation 1456 which provides models and test coverage for non-deterministic programming as taught herein." Support for this amendment can be found in the Specification at page 2, lines 5-6. Similarly, a replacement Figure 14 has been filed with this amendment. Box 1456 in Replacement Figure 14 now reads "non-deterministic test coverage implementations." Support for this amendment can be found in the Specification at page 2, lines 5-6.

As Figure 14 illustrates implementations of the methods and systems taught within the application, applicants respectfully submit that Figure 14 has material that is not old, and so is not prior art. Applicants respectfully request removal of this objection.

35 U.S.C. § 101 Rejections of Claims 1-6

The Examiner rejects claims 1-6 and 15-20 on the grounds that the claims recites a software algorithm that does not produce a useful tangible output, thereby failing to provide a tangible and concrete result.

Claims 1-6.

Applicants respectfully disagree with the Examiner's characterization of the claims and relevant law, and believe that the claims in their previous state satisfied 35 U.S.C. § 101. Nevertheless, applicants have amended the above-rejected claims in an effort to expedite prosecution. Specifically the claim now recites: "storing a representation of the created strategies in computer memory." Thus, the representation of the created strategies are stored, providing a "useful, tangible output." Applicants respectfully submit that Claim 1, as amended, is directed to statutory subject matter and request that the rejection under 35 U.S.C. § 101 be withdrawn. Claims 2-6 depend on Claim 1 and at least for that reason should also not be subject to a 35 U.S.C. § 101 rejection.

Claims 15,16, and 18-20.

Applicants respectfully submit that claims 15-16 and 18-20 are currently statutory as they claim functional descriptive material stored on a computer-readable medium.

MPEP 2106 divides descriptive material into "functional descriptive material" – "data structures and computer programs which impart functionality when employed as a computer component" and "non-functional descriptive material" – music, literary works, and mere arrangements of data.

Functional descriptive material is statutory when stored on a computer-readable medium. *See Warmerdam*, 33 F.3d at 1360-61, 31 USPQ2d at 1759 (claim to computer having a specific data structure stored in memory held statutory product-by-process claim.) Also see *In re Lowry*, 32 U.S.P.Q.2d 1031 ("memory" with a "data structure" held statutory.) [*See* MPEP 2106.IV.B.1.a.]

The material of claim 15 describes instructions for modeling program behavior, for verifying program behavior, and so forth. As such it is "computer programs which impart functionality when employed as a computer component" i.e., functional descriptive material.

Furthermore, the material is stored on a computer-readable medium. As the claim is to functional descriptive material stored on a computer-readable medium, applicants respectfully submit that Claim 15, as amended, is directed to statutory subject matter and request that the rejection under 35 U.S.C. § 101 be withdrawn. Claims 16 and 18-20 depend on Claim 15 and at least for that reason should also not be subject to a 35 U.S.C. § 101 rejection.

35 U.S.C. § 112 First Paragraph Rejections of Claims 1-6

The Action rejects claims 1-6 under 35 U.S.C. § 112 ¶1 as failing to comply with the enablement requirement.

Claim 1

The Action rejects claim 1 under the grounds that "the strategies merely increase the probability of executing/reaching the discrete sequences not reached. Therefore, the strategies do not guarantee that the states will be executed ..." [Action, ¶10.]

Applicants respectfully disagree with the examiner's reasoning, but to move the application forward have amended the claim. Claim 1 has been amended to read, in part, "executing the program under test under test conditions *using the stored created strategies* that cause the program to *have a higher probability to* execute through states that correspond to the *untested program behavior.*" Thus, we believe that claim 1 is not subject to a § 112 rejection.

Claim 5.

The Action rejects claim 5 on the grounds that "the claim recites, in part 'the executing program is instrumented with executable code that verifies upon execution that a program state conforms to a state of the graph.'", [Action, ¶10.] However, the Examiner states that "upon execution" means that the program has finished executing, and so the program cannot execute further instructions. [*See* Action, ¶10.] Applicants respectfully disagree with the examiner's reasoning, but to move the application forward have amended the claim as shown below.

> The method of claim 1 wherein the executing program is instrumented with executable code that verifies <u>during</u> execution that a program state conforms to a state of the graph.

35 U.S.C. § 112 Second Paragraph Rejections of Claims 1-6

Claim 1, lack of antecedent basis.

Claim 1 is said to lack antecedent basis for the phrase "the program under test". [Action, ¶ 11.]

The claim now reads in part:

> executing the program under test under test conditions using the stored created strategies that cause the program to have a higher probability to execute through states that correspond to the untested program behavior. [Emphasis added.]

The phrase "the program under test" now has sufficient antecedent basis. Applicants respectfully requests that the rejection be removed.

Claim 13, lack of antecedent basis.

Claim 13 is said to lack antecedent basis for the phrase "the created strategies." [Action, ¶ 11.] Applicants respectfully disagree. Claim 13 depends from claim 7. Claim 7 has the language "a strategy calculation program for creating strategies more likely to reach the untouched discrete sequences..." [Emphasis added.] Thus, there is sufficient antecedent basis for the contested claim 13 language.

Claim 5, the use of the term "instrumented."

The term "instrumented," used in claim 5 is said to be indefinite. [Action, ¶ 12.] The word "instrumented" is used in the application as shown below.

> For example, by examining the state space of a program, it can be determined whether or not the program performs as expected. In one example, an abstract state machine is used to create inputs to a program under test. In another example, an abstract state machine is run in parallel with an application in order to verify that a program performs according to behavior expected by the states of an abstract state machine. In another example, a program is instrumented in a testing environment where its program behavior is tested according to an executable specification comprising an abstract state machine. [Specification, p. 6, lines 1-8.]

> In such an example, developer or test teams assign probabilities to edges exiting a choice point. In other cases, an application is instrumented with code for counting exits from choice points. In such an example, choice points are assigned

> according to the distribution of counted exits from choice points. In another example, the probabilities of edges exiting a choice point are known based on the underlying nature of an application or prior known behavior of an application, or class of applications. [Specification, p. 24, lines 18-23.]

Further, "instrumentation" is a term of art known to those of ordinary skill in the art as shown by, e.g., the "Microsoft Enterprise Instrumentation Framework", as published on October 26, 2003, and available at the website http://www.microsoft.com/downloads/details.aspx?familyid=80df04bc-267d-4919-8bb4-1f84b7eb1368&displaylang=en. Applicants believe the term "instrumented" is sufficiently described to meet the requirements of §112 and respectfully request that the rejection be removed.

Claims 1-6, omitting essential steps

The Action rejects claims 1-6 under 35 U.S.C. § 112 ¶2 for omitting essential steps. [Action, ¶ 13.] Specifically, the step "establishing conditions that cause the program to execute through the states" is said to be missing.

Applicants disagree, but to more quickly move the claims toward patentability have amended the claims to recite, in part:

> ...executing the program under test under test conditions using the stored created strategies that cause the program to have a higher probability to execute through states that correspond to the untested program behavior....

Applicants believe the claim is now no longer subject to a 112 rejection and respectfully request that the rejection be removed.

Other, unspecified rejections.

The Examiner mentions that errors similar to those given in ¶¶ 11-13 are merely exemplary and that other, similar errors should be corrected. Applicant has attempted to do so. If the Examiner finds other errors, applicants will gladly correct them.

35 U.S.C. § 102 Rejections of claims 1-20

The Action rejects claims 1-20 under 35 U.S.C. 102(b) as being anticipated by U.S.

Patent No. 5,659,555 to Lee et al. (Lee). Applicants respectfully submit the claims are allowable over the cited art. For a 102(b) rejection to be proper, the cited art must show each and every element as set forth in a claim. (*See* MPEP § 2131.01.) However, the cited art does not describe each and every element. Accordingly, applicants request that all rejections be withdrawn.

Independent claim 1.

Amended independent claim 1 recites:

> A computerized method of creating test coverage for non-deterministic programs within a testing environment comprising:
> receiving a graph of edges and states representing a program under test;
> creating a continuous cycle of edges through the graph that reaches each edge in the graph at least once;
> splitting the continuous cycle into discrete sequences that end at edges reaching non-deterministic states uncontrollable by the testing environment in the graph;
> executing the program;
> determining untested program behavior as discrete sequences not reached by the program;
> creating strategies through the graph that have a higher probability of reaching discrete sequences not reached by the program;
> storing a representation of the created strategies in computer memory; and
> executing the program under test under test conditions using the stored created strategies that cause the program to have a higher probability to execute through states that correspond to the untested program behavior.

Lee neither teaches nor suggests "*creating a continuous cycle of edges through the graph that reaches each edge in the graph at least once.*"

Lee explicitly states "Because we do not do a reachability computation, we cannot, of course say absolutely whether a current global state s_i can be reached from the previous one s_{i-1}." [Lee, 11:20-23.] A reachability computation involves, at a minimum, creating a cycle of edges through the graph that reach each edge in the graph at least once. Then, it can be determined if the global state s_i can be reached from the previous one s_{i-1} by examining the cycle of edges through the graph. If a reachability computation is not performed, then "*creating a continuous cycle of edges through the graph that reaches each edge in the graph at least once*" i.e., a step toward a reachability computation, is explicitly not performed. Lee, further, teaches away from such a continuous cycle. A reachability computation is not done to "avoid exploding the states."

[Lee, 11:18-19.] Thus, Lee would not "*creat[e] a continuous cycle of edges through the graph that reaches each edge in the graph at least once*" to avoid the state explosion problem.

As further, explicit, teaching away, Lee states: "Computation and traversal of the global FSM are intractable for most real-life protocols. [Lee, 7:9-11.]

As Lee does not teach or suggest, at a minimum, "*creating a continuous cycle of edges through the graph that reaches each edge in the graph at least once*" claim 1 is not subject to a 102(b) rejection; applicants request that the rejection be withdrawn.

Claims 2-6.

Claims 2-6 ultimately depend on claim 1. Thus, at least for the reasons set forth above with respect to claim 1, claims 2-6 should also be in condition for allowance. These claims also set forth independently patentable combinations of method acts.

Independent Claim 7.

Independent claim 7 recites:

> A computer system comprising:
> memory and a central processing unit executing,
> a compiler for compiling an executable specification into an abstract state machine,
> a graphing program for creating a continuous cycle touching all edges of the abstract state machine, and for splitting the continuous cycle into discrete sequences that end at non-deterministic states;
> a strategy calculation program for creating strategies more likely to reach the untouched discrete sequences;
> a coverage program for executing a program and verifying that the program executes states corresponding to those modeled by discrete sequences of the abstract state machine and for determining untouched discrete sequences and for executing the program according to the created strategies and verifying whether the program executes states corresponding to the untouched discrete sequences.

Lee neither teaches nor suggests "*a graphing program for creating a continuous cycle touching all edges of the abstract state machine, and for splitting the continuous cycle into discrete sequences that end at non-deterministic states.*" Applicants will not belabor the point, but using the same reasoning as found in the discussion of claim 1, it can be seen that Lee does

not teach "a graphing program for creating a continuous cycle touching all edges of the abstract state machine" as a reachability analysis (which requires reaching each reachable edge in the graph at least once) is not performed, and strongly teaches against such claim language, to avoid the state explosion problem. [Lee, 11:18-23.] Thus claim 7 is in condition for allowance.

Claims 8-14.

Claims 8-14 ultimately depend on claim 7. Thus, at least for the reasons set forth above with respect to claim 7, claims 8-14 should also be in condition for allowance. These claims also set forth independently patentable system combinations.

Independent Claim 15.

Amended independent claim 15 recites:

> A computer-readable medium having thereon computer-executable instructions comprising:
> instructions for creating a model of program behavior comprising an abstract state machine with edge transitions;
> instructions for verifying program behavior;
> instructions for splitting the model of program behavior into sequences of at least two edge transitions ending at non-deterministic behavior;
> instructions for determining strategies for the sequences more likely to reach an identified program behavior; and
> instructions for causing a program to execute behavior corresponding to strategies for the sequences more likely to reach the identified program behavior.

Lee does not teach or suggest the claim language "instructions for creating a model of program behavior comprising an abstract state machine with edge transitions; instructions for splitting a the model of program behavior into sequences of at least two edge transitions ending at non-deterministic behavior."

In Lee, programs are modeled by using a random walk where the next input is chosen by surveying the possible inputs and then choosing the one that has not yet been "well tested." If more than one input falls in this category, then one is randomly chosen as an input. [*See* Lee, 8:37-9:32.] That is, in Lee, there is no two or more step look-ahead. Each input is chosen by looking just one step ahead. This is different from, and teaches away from "instructions for

splitting the model of program behavior into sequences of *at least two edge transitions* ending at non-deterministic behavior."

Further, the test input sequences in Lee end in two situations: first, "when the resulting output does not conform to the specification", and second "until the protocol implementation has been sufficiently tested." [Lee, 8:46-49.] Neither of these mentions non-deterministic behavior as a reason to *split the model of program behavior into sequences.* Further, this teaches away from instructions for *splitting the model of program behavior into sequences of at least two edge transitions ending at non-deterministic behavior*; as the output at a non-deterministic sequence is, by definition not defined by the specification and so cannot conform to it, in the first case, and has nothing to do with the protocol implementation being sufficiently tested, as in the second case.

Thus, claim 15 is in condition for allowance.

Claims 16, and 18-20.

Claims 16, and 18-20 ultimately depend on claim 15. Thus, at least for the reasons set forth above with respect to claim 15, claims 16, and 18-20 should also be in condition for allowance. These claims also set forth independently patentable combinations.

Request for Interview

If any issues remain, the Examiner is formally requested to contact the undersigned attorney prior to issuance of the next Office action in order to arrange a telephonic interview. It is believed that a brief discussion of the merits of the present application may expedite prosecution. Applicants submit the foregoing formal Amendment so that the Examiner may fully evaluate Applicants' position, thereby enabling the interview to be more focused.

This request is being submitted under MPEP § 713.01, which indicates that an interview may be arranged in advance by a written request.

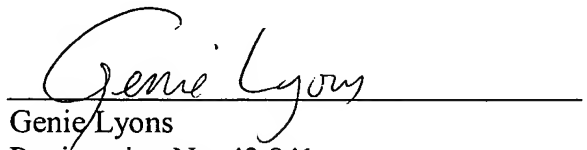
Conclusion

The claims should be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

By 
Genie Lyons
Registration No. 43,841